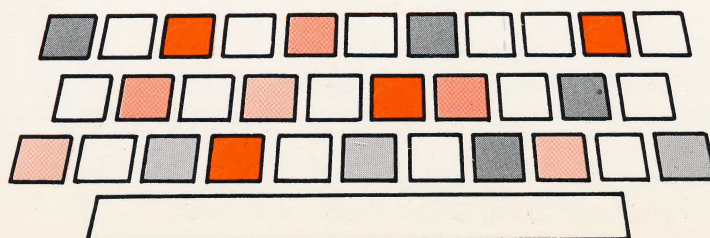
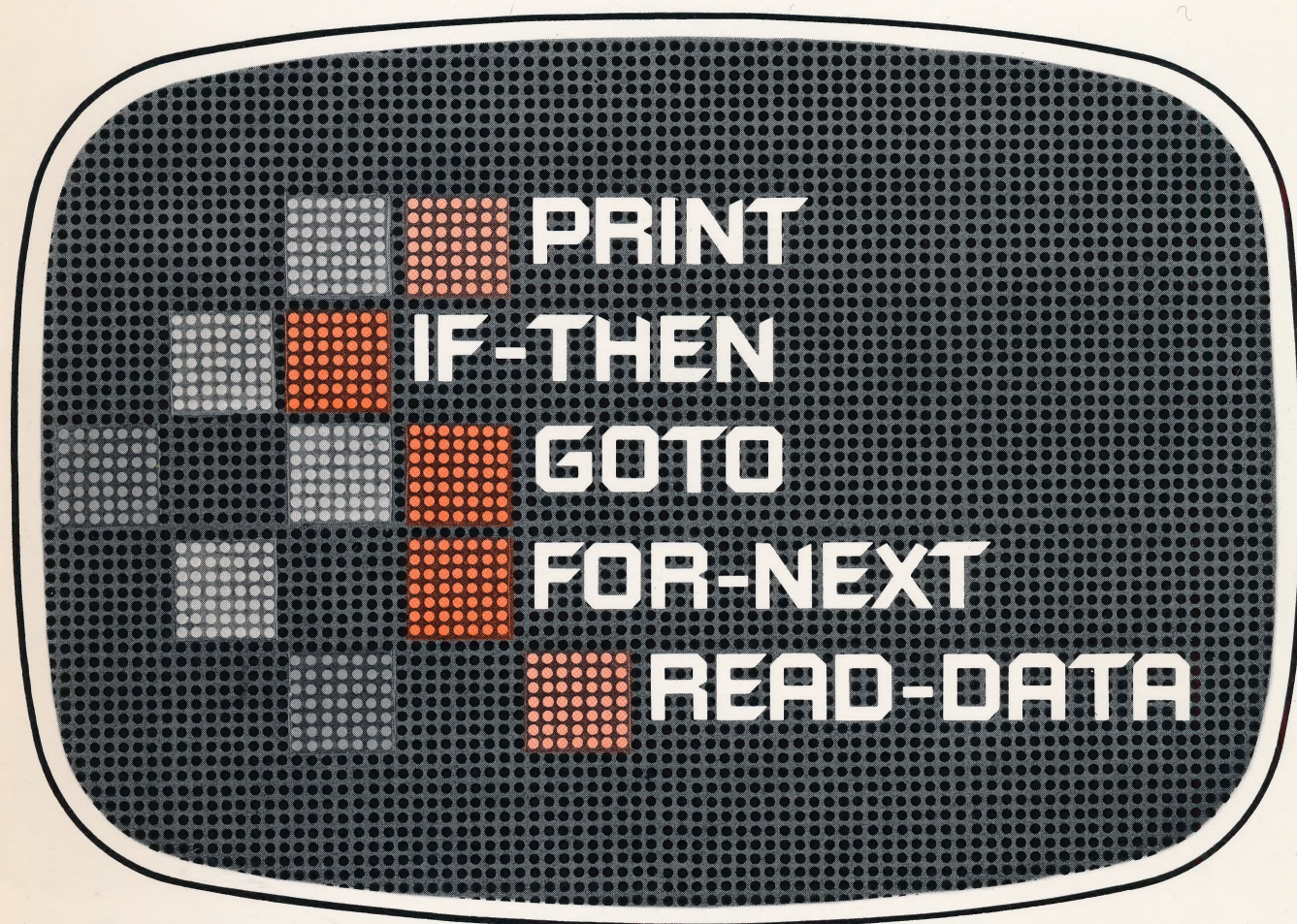


RANDOM HOUSE

PRACTICING PROGRAMMING ON THE ATARI



Written by **Maryann Marrapodi**
Howard Roger Budin
Mariano Guzman

Project Editor: Sandra Kelley
Text Design and Production by Merlin Communications, Inc.
Cover Design and Production by Thomas Vroman Associates, Inc.

Atari is the registered trademark of Atari, Inc.

© 1984 by Random House, Inc.
All rights reserved under International and Pan-American Copyright
Conventions. Published in the United States by Random House, Inc.,
New York.
International Standard Book Number: 0-676-39237-7
Manufactured in the United States of America

Random House **PRACTICING PROGRAMMING ON THE ATARI**



Random House School Division

STEPS FOR PRACTICING PROGRAMMING ON THE ATARI

1. Find the lesson you are going to do.
2. Read the lesson explanation carefully.
3. Do the **Writing Output** section of the lesson.
Write the missing output in the space provided.
4. Do the **Writing Input** section of the lesson.
Write the missing input on the blank lines.
5. Do the **Problem Solving** section of the lesson. When you write your own program, use as many of the blank lines as you need. If you need more lines, use a separate piece of paper.
6. Check your answers for each section by keyboarding each program into your Atari computer and RUNning the completed program.

Written by Margaret Sharpwood
Howard Roger Bunker
Marlene Quinter

Project Editor: Sandra Kufky

Text Design and Production by Martin Communications, Inc.

Cover Design and Production by Thomas Aronson Associates, Inc.

Atari is the registered trademark of Atari, Inc.

© 1984 by Spectrum Books, Inc.

All rights reserved under copyright and Patent Laws, Copyright

Compliance. Published in the United States by Spectrum Books, Inc.

New York

Random House Children's Division
Manufactured in the United States of America

TABLE OF CONTENTS

Steps for Practicing Programming on the ATARI	2
Lesson 1 Using the PRINT Statement	4
Lesson 2 Using Arithmetic Operations	7
Lesson 3 Printing Character Graphics	10
Lesson 4 Using Punctuation Marks	13
Lesson 5 Review	16
Lesson 6 Using LET Statements	18
Lesson 7 Using String Variables	21
Lesson 8 Using INPUT Statements	24
Lesson 9 Using IF-THEN Statements	27
Lesson 10 Review	30
Lesson 11 Using GOTO Statements	32
Lesson 12 Using Random and Integer Functions	35
Lesson 13 Using the FOR-NEXT Loop	38
Lesson 14 More About Loops	41
Lesson 15 Review	44
Lesson 16 Using READ-DATA Statements	46
Lesson 17 Producing Low-Resolution Graphics	49
Lesson 18 Animating Graphics	52
Lesson 19 Using String Functions	55
Lesson 20 Review	58
Summary of BASIC Statements and Commands for the ATARI	60
Glossary	62
Overview of Disk and Tape Commands	63
Graphics Sheet for the Atari Computer	64

LESSON 1:

Using the PRINT Statement

■ You can use the PRINT statement to display any keyboard character on the screen. For example, you can keyboard

```
PRINT "I AM AN ATARI"
```

■ Then you press the RETURN key. The computer will display the characters I AM AN ATARI, and the cursor will move to the next line on the screen.

■ When you write a program with more than one numbered line, you need to press RETURN after each line. Then type RUN to make the computer display your message. For example,

```
10 PRINT "HELLO"  
20 END  
READY  
[ ] RUN  
HELLO
```

□□□□□□□□□□□□□□□□□□□□□□□□ Writing Output □□□□□□□□□□□□□□□□□□□□□□□□

1. Read the program. Then look at the output and write what is missing on the blank lines.

```
10 PRINT "I LIKE WORKING"  
20 PRINT "WITH AN"  
30 PRINT "ATARI COMPUTER."  
40 END
```

```
[ ] RUN  
I LIKE WORKING  
_____  
_____
```

2. Read the program. Then look at the output and write what is missing on the blank lines.
NOTE: To leave a blank line on the screen, you use the PRINT statement followed by pressing RETURN.

```
10 PRINT "I CAN SPEAK MANY  
LANGUAGES."  
20 PRINT  
30 PRINT "MY FAVORITE IS BASIC."  
40 PRINT  
50 PRINT "WHICH IS YOUR FAVORITE  
LANGUAGE?"  
60 END
```

```
[ ] RUN  
_____  
_____  
MY FAVORITE IS BASIC.  
_____  
_____
```


3. Look at the output on the screen. Then read the program and fill in the missing lines.

```
[ ] RUN
HUMANS UNDERSTAND COMPUTERS.

CAN COMPUTERS UNDERSTAND HUMANS?
```

10 _____
 20 PRINT _____
 30 _____
 40 END

4. Look at the output on the screen. Then read the program and fill in the missing lines.

```
[ ] RUN
A
T
A
R
I
```

10 _____
 20 _____
 30 PRINT "A"
 40 _____
 50 _____
 60 END

5. Read the output on the screen. Then write a program which will produce this output.
 HINT: Use spaces inside the quotation marks to display the output as shown.

```
[ ] RUN
THIS PROGRAM HAS NO BAGS.

    THIS PROGRAM HAS NO BAGS.

        THIS PROGRAM HAS NO BAGS.

OOPS! I MEAN BUGS!
```

10 _____
 20 _____
 30 _____
 40 _____
 50 _____
 60 _____
 70 _____
 80 _____

6. Read the output on the screen. Then write a program which will produce this output.

[]RUN

THESE WORDS ARE PALINDROMES:

LEVEL

NOON

MADAM

PEEP

MOM

10 _____

20 _____

30 _____

40 _____

50 _____

60 _____

70 _____

80 _____

90 _____

100 _____

110 _____

120 _____

Problem Solving

7. Write a program which will display your name in the center of the screen. Print your address beneath it.
8. Write a program which will display the name of your school on the first line, and the names of five teachers on the next five lines.

[illegible]

THE PROGRAM HAS NO BIAS

LESSON 2:

Using Arithmetic Operations

■ You can use the PRINT statement to have the computer calculate and print the answer to an arithmetic problem. For example, when you key-board

```
10 PRINT 7*6
```

the output that is displayed is 42. Do not use quotation marks when you want the computer to do a calculation and only print the answer.

Writing Output

1. Read the program. Then look at the output and write what is missing on the blank lines.
NOTE: The symbol * means "multiply" and / means "divide."

```
10 PRINT 15*3
20 PRINT 15+3
30 PRINT 15/3
40 PRINT 15-3
50 END
```

[] RUN

5

2. Read the program. Then look at the output and write what is missing on the blank lines.

```
10 PRINT "10 PLUS 2 IS "
20 PRINT 10 + 2
30 PRINT "10 MINUS 2 IS "
40 PRINT 10 - 2
50 PRINT "10 DIVIDED BY 2 IS"
60 PRINT 10/2
70 PRINT "10 MULTIPLIED BY 2 IS"
80 PRINT 10*2
90 END
```

[] RUN

10 PLUS 2 IS

10 DIVIDED BY 2 IS

3. Look at the output on the screen. Then read the program and fill in the missing lines.

```
[ ] RUN
THE SUM OF 13 AND 87 IS
100

120 DIVIDED BY 12 IS
10

25 MULTIPLIED BY 8 IS
200
```

10 PRINT "THE SUM OF 13 AND 87 IS"

20 _____

30 PRINT

40 _____

50 _____

60 PRINT

70 _____

80 PRINT 25*8

90 END

4. Look at the output on the screen. Then read the program and fill in the missing lines.

```
[ ] RUN
MAGIC NINES.

THE SUM OF THE DIGITS OF
THE PRODUCT IS NINE.

9 X 2 =
18
1 + 8 =
9

9 X 5 =
45
4 + 5 =
9
```

10 _____

20 PRINT

30 PRINT "THE SUM OF THE DIGITS OF"

40 _____

50 PRINT

60 _____

70 PRINT 9*2

80 _____

90 PRINT 1+8

100 PRINT

110 _____

120 PRINT 9*5

130 _____

140 _____

150 _____

5. Read the output on the screen. Write a program that will produce this output.

[]RUN

ADDING THREE ODD NUMBERS
WILL PRODUCE AN ODD SUM.

$$1 + 3 + 5 = 9$$
$$21 + 33 + 45 = 99$$
$$101 + 153 + 125 = 379$$
[illegible]

Problem Solving

6. Write a program which uses at least three examples to show that the sum of any two odd numbers is an even number. Include a title for your program.
7. Write a program which will calculate and print the sums for five decimal numbers. Print each problem on a separate line. Print its sum beneath it.

This image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page. There are approximately 20 lines visible. The paper has a slightly off-white or cream color. There are no margins, text, or other markings on the page.[illegible]

LESSON 3:

Printing Character Graphics

■ You can use symbols and spaces to draw a picture. The Atari computer allows you to place 38 symbols horizontally and 24 vertically. For example, if you keyboard

```
10 PRINT "*" : *
```

the screen will display an asterisk, a colon, and another asterisk.

Writing Output

1. Read the program. Then look at the output and write what is missing on the blank lines.

```

10 PRINT "ROSIE ROBOT."
20 PRINT
30 PRINT "*****"
40 PRINT " *      * "
50 PRINT " *    ^    * "
60 PRINT " *    @    * "
70 PRINT "*****"
80 END

```

[] RUN
ROSIE ROBOT.

* * *

* * * * *

2. Read the program. Then look at the output and write what is missing on the blank lines.

```
10 PRINT "A LARGE T!"
20 PRINT
30 PRINT "TTTTTTTTT"
40 PRINT "    T "
50 PRINT "    T "
60 PRINT "    T "
70 PRINT "    T "
80 END
```

[] RUN
A LARGE T!

T

T

Writing Input

3. Look at the output on the screen. Then read the program and fill in the missing lines.

[] RUN
HERE IS A RECTANGLE.

* * * * *

 * * * * *

```

10 _____
20 PRINT
30 _____
40 PRINT ":":::""
50 _____
60 _____
70 END

```

4. Look at the output on the screen. Then read the program and fill in the missing lines.

```
[ ] RUN
```

WAVES

[illegible]

5. Look at the output on the screen. Then read the program and fill in the missing lines.

[] RUN
ROSIE'S BROTHER.

```

!!!!!!!!!!!!!!!!
|
|  @  @
|   ^
|  _
|
!!!!!!!!!!!!!!!!
  ||
  ||
<< || >>

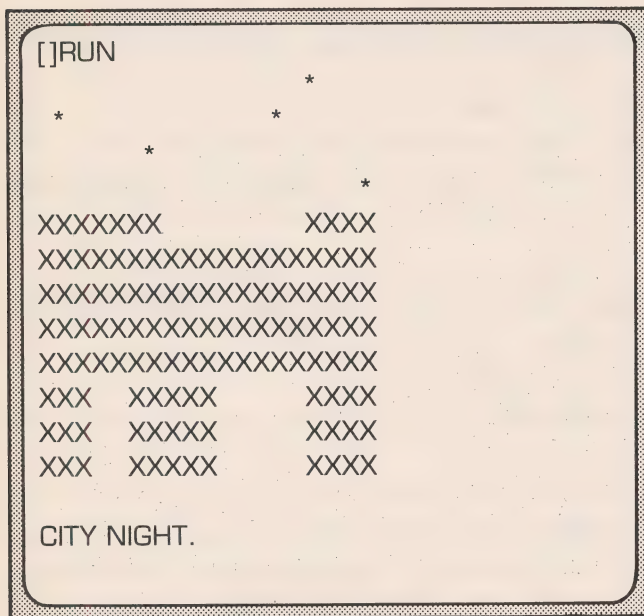
```

```

10 _____
20 PRINT _____
30 _____
40 PRINT "I      I"
50 _____
60 _____
70 _____
80 PRINT "!!!!!!!!!!"
90 _____
100 PRINT "  |  |  "
110 _____
120 END

```


6. Look at the output on the screen. Write a program that will display this output.



10 _____

20 _____

30 _____

40 _____

50 _____

60 _____

70 _____

80 _____

90 _____

100 _____

110 _____

120 _____

130 _____

140 _____

150 _____

Problem Solving

7. Write a program which uses spaces and symbols to display a picture of a sailboat.

This image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

8. Write a program which uses spaces and symbols to display a picture of a city skyline. Include at least four buildings in your picture.

LESSON 4:

Using Punctuation Marks

■ You can use a semicolon (;) to continue the output on the same line.

For example,

```
20 PRINT "ATARI ";
30 PRINT "COMPUTER"
```

displays the words ATARI COMPUTER on one line.

■ You can use a comma (,) to separate the output into four zones across the screen at space 0, space 10, space 20, and space 30. For example,

```
30 PRINT "I", "LIKE", "MY", "ATARI"
```

displays the words I LIKE MY ATARI across the screen.

□□□□□□□□□□□□□□□□□□□□□□□□ Writing Output □□□□□□□□□□□□□□□□□□□□□□□□

1. Read the program. Then look at the output and write what is missing on the blank lines.

```
10 PRINT "THE PRIMARY COLORS ARE:"
20 PRINT "RED", "YELLOW", "BLUE."
30 PRINT
40 PRINT "THE SECONDARY COLORS ARE:"
50 PRINT "GREEN ";
60 PRINT "ORANGE ";
70 PRINT "AND PURPLE."
80 END
```

[] RUN

THE PRIMARY COLORS ARE:

THE SECONDARY COLORS ARE:

-
2. Read the program. Then look at the output and write what is missing on the blank lines.

```
10 PRINT "ARITHMETIC FACTS"
20 PRINT
30 PRINT "9 + 9 = ";
40 PRINT 9+9
50 PRINT
60 PRINT "18 - 9 = ";
70 PRINT 18-9
80 PRINT
90 PRINT "18 / 9 = ";
100 PRINT 18/9
110 END
```

[] RUN

9 + 9 = 18

3. Look at the output on the screen. Then read the program and fill in the missing lines.

```
[ ] RUN
AUSTIN IS THE CAPITAL OF TEXAS.

OTHER CITIES IN TEXAS ARE:
HOUSTON  AND  DALLAS.
```

10 _____
 20 PRINT "IS THE CAPITAL OF TEXAS."
 30 _____
 40 PRINT "OTHER CITIES IN TEXAS ARE:"
 50 _____
 60 _____

4. Look at the output on the screen. Then read the program and fill in the missing lines.

```
[ ] RUN
BASE 10  BASE 2

1         1
2         10
3         11
4         100
5         101
```

10 PRINT "BASE 10", "BASE 2"
 20 _____
 30 _____
 40 _____
 50 _____
 60 _____
 70 PRINT "5", "101"
 80 _____

5. Look at the output on the screen. Then write a program that will produce this output.

```
[ ] RUN
TEAM 1  TEAM 2  TEAM 3  TEAM 4

ALICE   MARIA   BETTY   GINA
JOHN    JASON    BILL    TIM
SARA    LISA     ELLEN   ANN
TONY    MARIO    ALEX    BEN
```

10 _____
 20 _____
 30 _____
 40 _____
 50 _____
 60 _____
 70 _____

6. Look at the output on the screen. Then write a program that will produce this output.

[] RUN
SOME FRUITS ARE:
APPLES CHERRIES MELONS PEARS.

SOME VEGETABLES ARE:
SPINACH,
CARROTS,
BROCCOLI,
AND PEAS.

SOME GRAINS ARE:
RICE, OATS, WHEAT, AND BARLEY.

This image shows a single page from a lined notebook. The page features horizontal blue ruling lines spaced evenly down its length. On the left side, the numbers 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, and 120 are printed vertically, corresponding to each set of lines. The paper has a light cream or off-white tint. There is no handwriting or other markings on the page.

Problem Solving

7. Write a program which prints the names of five colors on one line of the screen. Each color name should be in a separate PRINT statement. Display a title for the output.
8. Write a program which prints the title COUNTING BY FOURS on the top line of the screen. Then print the numbers 4 to 48 by fours in four columns across the screen.

[illegible]

LESSON 5:

Review

Writing Output

1. Read the program. Then look at the output and write what is missing on the blank lines.

```
10 PRINT "WATCH HOW QUICKLY ";
20 PRINT "I DO MATH!"
30 PRINT
40 PRINT "163 X 3 = ";
50 PRINT 163*3
60 PRINT
70 PRINT "256 / 2 = ";
80 PRINT 256/2
90 PRINT
100 PRINT "203 + 694 + 726 = ";
110 PRINT 203+694+726
120 PRINT
130 PRINT "CAN YOU WORK ";
140 PRINT "AS QUICKLY?"
150 END
```

[]RUN

163 X 3 = 489

CAN YOU WORK AS QUICKLY?

Writing Input

2. Look at the output on the screen. Then read the program and fill in the missing lines.

NOTE: Press the key marked ^{CAPS}_{LOWR}

to print lower-case letters. Once this key has been pressed, use the SHIFT key to print capital letters. Remember to use all upper-case letters for statements and commands such as RUN and PRINT.

```
10 _____
20 PRINT
30 _____
40 _____
50 _____
60 _____
70 PRINT "SPURS", "100", "88", "94"
80 END
```

[]RUN				
BASKETBALL SCORES				
Team	Game 1	Game 2	Game 3	
ACES	123	98	104	
PROS	97	112	84	
STARS	67	119	95	
SPURS	100	88	94	

LESSON 6:

Using LET Statements

■ You can use a LET statement to assign a value to a variable and store it in a memory location. For example,

```
20 LET X = 3
```

stores the value 3 in a memory location named X. The computer will not display anything on the screen, however, unless you use a PRINT statement.

□□□□□□□□□□□□□□□□□□□□□□□□ Writing Output □□□□□□□□□□□□□□□□□□□□□□□□

1. Read the program. Then look at the output and write what is missing on the blank lines.

```
10 PRINT "PRINTING VARIABLES"
20 LET X = 1
30 LET Y = 2
40 LET Z = 3
50 PRINT "THE VALUE OF X IS ";
60 PRINT X
70 PRINT "THE VALUE OF Y IS ";
80 PRINT Y
90 PRINT "THE VALUE OF Z IS ";
100 PRINT Z
110 END
```

[] RUN

THE VALUE OF Z IS 3

-
2. Read the program. Then look at the output and write what is missing on the blank lines.

```
10 LET A = 2
20 PRINT "A = "; A
30 LET A = A * 2
40 PRINT "A = "; A
50 LET A = A * 2
60 PRINT "A = "; A
70 END
```

[] RUN

A = 4

3. Look at the output on the screen. Then read the program and fill in the missing lines.
NOTE: You can use a colon (:) to link, or concatenate, two statements on one line.

```
[ ] RUN
1
2
3
4
5
```

10 LET X = 1: PRINT X

20 _____

30 _____

40 LET X = X + 1: PRINT X

50 _____

60 _____

4. Read the output on the screen. Then read the program and fill in the missing lines.

```
[ ] RUN
CHANGING THE VALUES OF VARIABLES
```

A	B	C	D
1	2	3	4
2	4	6	8
6	12	18	24

10 LET A = 1: LET B = 2: LET C = 3:
LET D = 4

20 _____

30 _____

40 PRINT "A", "B", "C", "D"

50 _____

60 _____

70 LET A = A*2: LET B = B*2:
LET C = C*2: LET D = D*2

80 _____

90 _____

100 _____

110 END

5. Read the output on the screen. Then write a program that will produce this output. Use variables in your program.

```
[ ] RUN
VARIABLE      VARIABLE X 10
1              10
10             100
100            1000
```

10 _____

20 _____

30 _____

40 _____

50 _____

60 _____

6. Read the output on the screen. Then write a program that will produce this output. HINT: Use $N*N$ for the square and $N*N*N$ for the cube.

[]RUN Number	Square	Cube
1	1	1
2	4	8
3	9	27
4	16	64
5	25	125

A vertical number line with horizontal tick marks. The numbers 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 are written to the left of the line. The line itself is a single vertical line with horizontal segments connecting the tick marks.

Problem Solving

7. Write a program that will store the values 10, 20, and 30 as A, B, and C. Print these values in one column. Next to each, print the value of the number multiplied by 100.

This image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page. There are approximately 20 lines visible. The paper appears slightly aged or off-white. There is no handwriting or other markings on the page.

8. Write a program that will display any number, its value multiplied by 5, and its value multiplied by 10. Print your results in three columns.

This image shows a single page of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page. There are approximately 20 lines visible. The paper appears slightly aged or off-white. There is no handwriting or other markings on the page.

LESSON 7:

Using String Variables

■ You can use string variables to store characters or words in the computer's memory. For example,

```
20 LET N$ = "JENNY"
```

stores the word JENNY in a memory location named N\$. Be sure to put the word inside quotes and to use a string sign (\$) after the variable name.

■ To reserve space for the strings, start your program with a statement like

```
10 DIM N$(20)
```

DIM stands for "dimension." The DIM statement will reserve enough room in the computer's memory for as many as 20 characters in the string called N\$.

Writing Output

1. Read the program. Then look at the output and write what is missing on the blank lines.

[] RUN

```
15 DIM X$(20), Y$(20)
20 LET X$ = "COMPUTER IS FUN."
30 LET Y$ = "AN ATARI "
40 PRINT Y$
50 PRINT X$
60 PRINT Y$;X$
70 END
```

COMPUTER IS FUN.

2. Read the program. Then look at the output and write what is missing on the blank lines.
NOTE: Strings can be added together by using a semicolon.

[] RUN

```
5 DIM A$(5), B$(5), C$(5), D$(5)
10 LET A$ = "CALL"
20 LET B$ = "VIEW"
30 LET C$ = "VEAL"
40 LET D$ = "RE"
50 PRINT D$; A$
60 PRINT D$; C$
70 PRINT D$; B$
80 PRINT A$,B$,C$,D$
90 END
```

REVEAL

3. Look at the output on the screen. Then read the program and fill in the missing lines.

```
[ ] RUN
STRINGS CAN BE PRINTED IN ANY ORDER
IN ANY ORDER STRINGS CAN BE PRINTED
```

```
5 DIM X$(15), Y$(15), Z$(15)
10 LET X$ = " IN ANY ORDER "
20 LET Y$ = " STRINGS CAN "
30 LET Z$ = " BE PRINTED "

40 _____
50 _____
60 _____
```

4. Read the output on the screen. Then read the program and fill in the missing lines.

```
[ ] RUN
LIU, C.      ASTOR, B.
GREEN, L.    BRADY, G.
DIAZ, R.

THE NAMES IN ALPHABETICAL ORDER ARE:
ASTOR, B.
BRADY, G.
DIAZ, R.
GREEN, L.
LIU, C.
```

```
5 _____
10 LET A$ = "LIU, C.":LET B$ = "ASTOR, B."
20 _____
30 LET E$ = "DIAZ, R."
40 _____
50 PRINT C$, D$
60 _____
70 PRINT
80 _____
90 _____
100 PRINT D$
110 _____
120 _____
130 _____
140 END
```

5. Read the output on the screen. Then write a program that will produce this output.

```
[ ] RUN
MID    DAY    NIGHT    LIGHT
MIDNIGHT
DAYLIGHT
```

```
10 _____
20 _____
30 _____
40 _____
50 _____
60 _____
70 _____
```


6. Read the output on the screen. Then write a program that will produce this output. HINT: Use only four string variables. Concatenate the variables on two lines.

[] RUN

stpo
tops
post
spot
stop
pots

10

20

30

40

50

60

70

80

90

100

Problem Solving

7. Write a program that will store four different names as four string variables. Print these names in five different ways on your screen.

[illegible]

8. Write a program that uses four string variables to store four different letters. Print these letters as follows: in the four print zones; next to each other; and on four different lines.

[illegible]

LESSON 8:

Using INPUT Statements

■ You can use the INPUT statement to make the computer stop and wait for the person using the program to keyboard in information. For example,

```
10 DIM L$ [1]
20 PRINT "TYPE A LETTER"
30 INPUT L$
```

makes the computer display the words TYPE A LETTER and places a question mark on the line below. The computer then waits until the user inputs the data and presses the RETURN key before executing the rest of the program.

Writing Output

1. Read the program. Then look at the output and write what is missing on the blank lines.

```
10 DIM N$ [15]
20 PRINT "WHAT IS YOUR NAME?"
30 INPUT N$
40 PRINT "HELLO, ";N$
50 PRINT "MY NAME IS ATARI."
60 PRINT "IT'S NICE TO MEET YOU."
70 END
```

```
[ ]RUN
WHAT IS YOUR NAME?
```

HELLO, BRUCE

2. Read the program. Then look at the output and write what is missing on the blank lines.

```
10 PRINT "IN WHICH YEAR WERE YOU BORN?"
20 INPUT Y
30 LET Z = 2000 - Y
40 PRINT "IN THE YEAR 2000, YOU WILL BE "
50 PRINT Z; " YEARS OLD. "
60 END
```

```
[ ]RUN
```

? 1965

IN THE YEAR 2000, YOU WILL BE

3. Look at the output on the screen. Then read the program and fill in the missing lines.

```
[ ] RUN
TYPE A NUMBER BETWEEN 1 AND 10.
?6
TYPE ANOTHER NUMBER FROM 1 TO 10.
?5
THE SUM OF THESE NUMBERS IS 11.
THEIR PRODUCT IS 30.
```

```
10 PRINT "TYPE A NUMBER BETWEEN
1 AND 10."
```

```
20 _____
```

```
30 _____
```

```
40 INPUT Y
```

```
50 PRINT "THE SUM OF THESE NUMBERS
IS ";X + Y; "."
```

```
60 _____
```

```
70 END
```

4. Look at the output on the screen. Then read the program and fill in the missing lines.

```
[ ] RUN
IN WHICH STATE WERE YOU BORN?
?COLORADO
COLORADO IS A NICE STATE.
I WAS BORN IN CALIFORNIA.
```

```
10 _____
```

```
20 PRINT "IN WHICH STATE WERE YOU
BORN?"
```

```
30 _____
```

```
40 PRINT $$; " IS A NICE STATE. "
```

```
50 _____
```

```
60 _____
```

5. Read the output on the screen. Then write a program that will produce this output.
HINT: Use the formula $A = L \times W$ to find the area of a rectangle.

```
[ ] RUN
I CAN FIND THE AREA OF RECTANGLES.
WHAT IS THE LENGTH?
?20
WHAT IS THE WIDTH?
?5
THE AREA IS 100.
```

```
10 _____
```

```
20 _____
```

```
30 _____
```

```
40 _____
```

```
50 _____
```

```
60 _____
```

```
70 _____
```

```
80 _____
```


6. Look at the output on the screen. Then write a program that will produce this output.

[] RUN
TYPE THE NAME OF AN ANIMAL.
? MONKEY
TYPE AN ADJECTIVE THAT BEGINS
WITH THE SAME LETTER AS THE
ANIMAL.
? MERRY

THE MERRY MONKEY MAKES ME SMILE.

10 _____

20 _____

30 _____

40 _____

50 _____

60 _____

70 _____

80 _____

90 _____

Problem Solving

7. Write a program which will allow the user to input any two numbers. Use these numbers to display the area of a triangle. The formula for the area of a triangle is $A = (B * H)/2$.
8. Write a program which allows the user to input five letters. Display these letters in reverse order on the screen.

This image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page. There are approximately 20 lines visible. The paper has a slightly textured appearance and some minor discoloration or faint smudges, particularly near the top edge. The overall tone is off-white or light cream.[illegible]

LESSON 9:

Using IF-THEN Statements

■ You can use an IF-THEN statement to tell the computer how to respond to different inputs. For example,

```
40 IF X = 3 THEN PRINT "THAT'S THE NUMBER!"
```

will tell the computer to display the words THAT'S THE NUMBER! only if the user inputs 3. If another number is entered, the computer will look for further instructions.

Writing Output

1. Read the program. Then look at the output and write what is missing on the blank lines.

```
10 DIM S$(10)
20 PRINT "IS $ THE SYMBOL FOR A STRING"
30 PRINT "OR FOR AN INPUT?"
40 INPUT S$
50 IF S$ = "STRING" THEN PRINT "THAT'S
RIGHT!"
60 IF S$ = "INPUT" THEN PRINT "NO.
$ IS THE SYMBOL FOR A STRING."
70 END
```

```
[ ]RUN
IS $ THE SYMBOL FOR A STRING
OR FOR AN INPUT?
?STRING
```

```
READY
[ ]RUN
IS $ THE SYMBOL FOR A STRING
OR FOR AN INPUT?
?INPUT
```

Writing Input

2. Look at the output on the screen. Then read the program and fill in the missing lines.

```
[ ]RUN
IS A MISTAKE IN A PROGRAM CALLED
A BUG OR A BUD?
?BUD
NO. A BUD IS A YOUNG FLOWER.

READY
[ ]RUN
IS A MISTAKE IN A PROGRAM CALLED
A BUG OR A BUD?
?BUG
I'LL BET YOUR PROGRAMS HAVE NO BUGS!
```

```
10 DIM R$(5)
20 _____
30 PRINT "A BUG OR A BUD?"
40 _____
50 _____
60 _____
70 _____
```


3. Look at the output on the screen. Then read the program and fill in the missing lines.

```
[ ]RUN
HOW MANY STATES ARE IN THE U.S.?
?48
NO. THAT'S TOO LOW.
```

```
READY
[ ]RUN
HOW MANY STATES ARE IN THE U.S.?
?60
NO. THAT'S TOO HIGH.
```

```
READY
[ ]RUN
HOW MANY STATES ARE IN THE U.S.?
?50
THAT'S EXACTLY RIGHT!
```

```
10 PRINT "HOW MANY STATES ARE IN THE
U.S.?"
```

```
20 INPUT N
```

```
30 _____
```

```
40 _____
```

```
50 _____
```

```
60 END
```

4. Read the output on the screen. Then write a program that will produce this output.

```
[ ]RUN
FOR HOW MANY YEARS IS A PRESIDENT
ELECTED TO OFFICE?
?8
NO. THAT'S TOO MANY YEARS.
```

```
READY
[ ]RUN
FOR HOW MANY YEARS IS A PRESIDENT
ELECTED TO OFFICE?
?3
NO. THAT'S TOO FEW YEARS.
```

```
READY
[ ]RUN
FOR HOW MANY YEARS IS A PRESIDENT
ELECTED TO OFFICE?
?4
YES!
```

```
10 _____
```

```
20 _____
```

```
30 _____
```

```
40 _____
```

```
_____
```

```
50 _____
```

```
_____
```

```
60 _____
```

```
70 _____
```

5. Read the output on the screen. Then write a program that will produce this output.

[J]RUN
IS A GROUP OF LIONS CALLED A
PRIDE OR A HERD?
?HERD
THAT'S NOT CORRECT.

READY

[] RUN
IS A GROUP OF LIONS CALLED A
PRIDE OR A HERD?
?PRIDE
THAT'S RIGHT.

A vertical number line with horizontal tick marks. The numbers 10, 20, 30, 40, 50, 60, and 70 are written to the left of the line, corresponding to each tick mark. The line itself is a single vertical line extending from the top of the 10 mark to the bottom of the 70 mark.

6. Read the output on the screen. Then write a program that will produce this output.

[] RUN
HOW MANY HOURS ARE IN A DAY?
?36
NO. THAT'S WRONG.

READY

```
[ ] RUN
HOW MANY HOURS ARE IN A DAY?
?24
THAT'S CORRECT!
```

10 _____

20 _____

30 _____

40 _____

50 _____

 Problem Solving

7. Write a program which asks the user to name the capital of the United States. Program the computer to tell the user whether his or her answer is correct or incorrect.

8. Write a program which asks the user the number of centimeters in a meter. Program the computer to tell the user whether his or her answer is too high, too low, or exactly right.

LESSON 10:

Review

Writing Output

1. Read the program. Then look at the output and write what is missing on the blank lines.

```
10 DIM N$(20)
20 PRINT "WHAT IS YOUR NAME?"
30 INPUT N$
40 PRINT "IN WHICH YEAR DID COLUMBUS
DISCOVER AMERICA?"
50 INPUT Y
60 IF Y = 1492 THEN PRINT "YOU ARE
RIGHT, " ; N$; "."
70 IF Y > 1492 THEN PRINT "THAT'S NOT
RIGHT, " ; N$; "."
80 IF Y < 1492 THEN PRINT " IT WAS LATER,
"; N$; "."
90 END
```

[] RUN

?BARRY

?1372

READY

[] RUN

?BARRY

?1695

READY

[] RUN

?BARRY

?1492

Writing Input

2. Look at the output on the screen. Then read the program and fill in the missing lines.

```
[ ] RUN
HOW MANY SIDES ARE IN A PENTAGON?
?5
THAT'S CORRECT.

READY
[ ] RUN
HOW MANY SIDES ARE IN A PENTAGON?
?4
THAT'S INCORRECT. THERE ARE FIVE SIDES.
```

10

20 INPUT S

30

40

50 END

3. Read the output on the screen. Then write a program that will produce this output. HINT: Don't forget the spaces between lines.

[] RUN

WHICH OF THESE IS NOT PART OF A MODERN COMPUTER?

CPU

INPUT DEVICE

OUTPUT DEVICE

VACUUM TUBES

?VACUUM TUBES

CORRECT. MODERN COMPUTERS USE SILICON CHIPS.

READY

[] RUN

WHICH OF THESE IS NOT PART OF A MODERN COMPUTER?

CPU

INPUT DEVICE

OUTPUT DEVICE

VACUUM TUBES

?CPU

WRONG. VACUUM TUBES IS THE ANSWER.

10 _____

20 _____

30 _____

40 _____

50 _____

60 _____

70 _____

80 _____

90 _____

100 _____

110 _____

120 _____

130 _____

Problem Solving



4. Write a program which asks the user to input a five-letter word, letter by letter. Have the computer print the word on one line, and print the word reversed on the second line.

5. Write a program which asks the user to input his or her height in inches. Then display the user's height in centimeters. HINT: One inch = 2.54 cm.

[illegible]

LESSON 11:

Using GOTO Statements

■ You can use GOTO with an IF—THEN statement to send the computer to another part of the program. For example,

```
50 IF C$ < > "YES" THEN GOTO 80
```

sends the computer to line 80 if C\$ is not equal to YES. If C\$ equals YES, the computer moves to the next line in the program.

Writing Output

1. Read the program. Then look at the output and write what is missing on the blank lines.
NOTE: You can use GOTO without IF to make the computer skip lines.

```
10 DIM R$ [15]
20 PRINT "CAN YOU GUESS WHERE I LIVE?"
30 PRINT "TYPE THE NAME OF A STATE."
40 INPUT R$
50 IF R$ < > "VERMONT" THEN GOTO 80
60 PRINT "YOU GUESSED IT!"
70 GOTO 90
80 PRINT "NO. I LIVE IN VERMONT."
90 END
```

```
[ ] RUN
CAN YOU GUESS WHERE I LIVE?
TYPE THE NAME OF A STATE.
?FLORIDA
```

```
READY
[ ] RUN
CAN YOU GUESS WHERE I LIVE?
TYPE THE NAME OF A STATE.
?VERMONT
```

-
2. Read the program. Then look at the output and write what is missing on the blank lines.

```
10 PRINT "WHAT IS THE PRODUCT OF
10 X 55?"
20 INPUT A
30 IF A = 550 THEN GOTO 60
40 PRINT "NO. TRY AGAIN."
50 GOTO 20
60 PRINT "THAT'S CORRECT."
70 END
```

```
[ ] RUN
WHAT IS THE PRODUCT OF 10 X 55?
?500
```

```
?550
```

3. Read the output on the screen. Then read the program and fill in the missing lines.

```
[ ] RUN
HOW MANY BYTES ARE IN A KILOBYTE?
?100
TOO FEW! TRY AGAIN.
?10000
TOO MANY! TRY AGAIN.
?1000
YOU REALLY KNOW COMPUTERS!
```

```
10 PRINT "HOW MANY BYTES ARE IN A
KILOBYTE?"
20 INPUT B

30 _____

40 _____

50 PRINT "YOU REALLY KNOW COM-
PUTERS!"

60 _____

70 PRINT "TOO MANY! TRY AGAIN."
80 GOTO 20
90 PRINT "TOO FEW! TRY AGAIN."

100 _____
110 END
```

4. Read the output on the screen. Write a program that will produce this output.

```
[ ] RUN
WHICH COUNTRY IS DIRECTLY NORTH
OF THE UNITED STATES?
?MEXICO
THAT'S NOT IT. TRY AGAIN.
WHICH COUNTRY IS DIRECTLY NORTH
OF THE UNITED STATES?
?CANADA
THAT'S IT!
```

```
10 _____

20 _____

30 _____

40 _____

50 _____

60 _____

70 _____

80 _____

90 _____
```


5. Read the output on the screen. Then write a program that will produce this output.

```
[ ] RUN
TYPE A NUMBER.
?4
TYPE ANOTHER NUMBER.
?45
WOULD YOU LIKE TO SEE THEIR SUM
OR THEIR PRODUCT?
?PRODUCT
THE PRODUCT OF 4 AND 45 IS 180.
```

```
READY
[ ] RUN
TYPE A NUMBER.
?50
TYPE ANOTHER NUMBER.
?125
WOULD YOU LIKE TO SEE THEIR SUM
OR THEIR PRODUCT?
?SUM
THE SUM OF 50 AND 125 IS 175.
```

10	
20	
30	
40	
50	
60	
70	
80	
90	
100	
110	
120	
130	

Problem Solving

6. Write a program which allows the user to input any number. Make the computer let the user know if his or her number is zero, less than zero, or greater than zero.

7. Write a program to test the user's knowledge of any state and its capital question. If his or her answer is correct, display a reward statement. If it is incorrect, inform the user and allow another try.

[illegible]

Using Random and Integer Functions

■ You can use the `RND(1)` function to have the computer pick a random decimal number between 0 and 1. For example,

```
30 PRINT RND[1]
```

might display the number 0.2341506989.

■ If you want a larger number, you can multiply the decimal by 10, which gives you 2.341506989. To get a simple whole number, you use the INT function. For example,

```
40 PRINT INT (2.341506989)
```

drops the numbers to the right of the decimal point and displays the integer 2. Multiplying a random number by 10 and printing the integer part of that number gives you the numbers 0-9. To get the numbers 1-10, you need to add a 1.

Writing Output

1. Read the program. Then look at the output and write what is missing on the blank lines.

```
10 LET X = RND[1]
20 PRINT X
30 PRINT 10 * X
40 PRINT 100 * X
50 PRINT INT[10 * X]
60 END
```

```
[ ]RUN
0.34562109
3.4562109
```

2. Read the program. Then look at the output and write what is missing on the blank lines.

```

5 DIM R$(3)
10 PRINT "HERE ARE TWO NUMBERS BE-
TWEEN 1 AND 10."
20 LET X = INT(10 * RND(1))
30 LET Y = INT(10 * RND(1))
40 PRINT X,Y
50 PRINT "THE SUM OF " ;X; " AND " ;Y;
" IS " ;X + Y; "."
60 PRINT "THE PRODUCT OF " ;X; " AND " ;
Y; " IS " ;X * Y; "."
70 PRINT "WOULD YOU LIKE TO SEE
MORE?"
80 INPUT R$
90 IF R$ = "NO" THEN GOTO 110
100 GOTO 10
110 END

```

```
[ ]RUN
HERE ARE TWO NUMBERS BETWEEN 1
AND 10.
```

3 7

WOULD YOU LIKE TO SEE MORE?
?NO

3. Look at the output on the screen. Then read the program and fill in the missing lines.

[] RUN
TRY TO GUESS MY NUMBER.
IT IS BETWEEN 0 AND 9.
TYPE A NUMBER.
?7
NO, THE NUMBER WAS 3.

```
10 LET V = INT(10 * RND(1))  
20 _____  
30 PRINT "IT IS BETWEEN 0 AND 9."  
40 _____  
50 _____  
60 IF R < V THEN GOTO 90  
70 PRINT "YOU GUESSED IT!"  
80 GOTO 100  
90 _____  
100 _____
```

4. Look at the output on the screen. Then read the program and fill in the missing lines.

```
[ ]RUN
CAN YOU GUESS A NUMBER LARGER
THAN MINE?
MY NUMBER IS BETWEEN 1 AND 100.
TYPE A NUMBER BETWEEN 1 AND 100.
?95
95 IS LARGER THAN 87! YOU WIN!
```

```
READY
[.]RUN
CAN YOU GUESS A NUMBER LARGER
THAN MINE?
MY NUMBER IS BETWEEN 1 AND 100.
TYPE A NUMBER BETWEEN 1 AND 100.
?78
78 IS SMALLER THAN 87! I WIN!
```

READY
[] RUN
CAN YOU GUESS A NUMBER LARGER
THAN MINE?
MY NUMBER IS BETWEEN 1 AND 100.
TYPE A NUMBER BETWEEN 1 AND 100.
?87
87 IS EQUAL TO 87! I WIN!

```

10 LET Z = INT(100 * RND(1)) + 1
20
30
40
50
60 INPUT N
70
80 IF N < Z THEN PRINT N; " IS
SMALLER THAN "; Z; " ! I WIN! "
90
100 END

```

5. Read the output on the screen. Then write a program that will produce this output.

[] RUN
THIS PROGRAM GIVES YOU PRACTICE
IN ADDING NUMBERS BETWEEN 1 AND 50.

42 + 35 =
?67
NO. TRY AGAIN.

42 + 35 =
?77
GOOD WORK!

10 _____

20 _____

30 _____

40 _____

50 _____

60 _____

70 _____

80 _____

90 _____

100 _____

110 _____

120 _____

CC BY-NC-ND 4.0

Problem Solving

6. Write a program which displays four random integer numbers in four columns. The numbers should be between 1 and 1000. Add the four numbers and display the sum.

[illegible]

7. Write a program which asks the user to guess a number between 1 and 50. If the number is correct, display a reward statement. If the number is too high or too low, let the user know and give him or her another chance.

LESSON 13:

Using the FOR-NEXT Loop

■ You can use the FOR—NEXT statements to tell the computer how many times to follow a set of commands. The FOR and NEXT statements always appear as a pair. For example,

```
10 FOR X = 1 TO 5
20 PRINT X
30 NEXT X
```

will display a list of numbers from 1 to 5.

Writing Output

1. Read the program. Then look at the output and write what is missing on the blank lines.

```
10 FOR N = 1 TO 5
20 PRINT "FIVE TIMES"
30 NEXT N
40 END
```

[] RUN

FIVE TIMES

2. Read the program. Then look at the output and write what is missing on the blank lines.

NOTE: If you want the numbers to print on one line, use the form PRINT C;" " ;.

```
10 FOR C = 1 TO 12
20 PRINT C;" " ;
30 NEXT C
40 END
```

[] RUN

Writing Input

3. Look at the output on the screen. Then read the program and fill in the missing lines.

```
[ ] RUN
NUMBER 10
NUMBER 11
NUMBER 12
NUMBER 13
NUMBER 14
NUMBER 15
```

```
10 _____
20 PRINT "NUMBER "; X
30 _____
40 END
```

4. Look at the output on the screen. Then read the program and fill in the missing lines.

```
[ ]RUN
WHAT IS YOUR NAME?
?CARMEN

HERE IS YOUR NAME PRINTED 10 TIMES:

CARMEN CARMEN CARMEN CARMEN
CARMEN CARMEN CARMEN CARMEN
CARMEN CARMEN
```

```
10 DIM N$(20)
20 _____
30 _____
40 PRINT
50 PRINT "HERE IS YOUR NAME PRINTED
10 TIMES:"
60 PRINT
70 _____
80 PRINT N$,
90 _____
100 END
```

5. Look at the output on the screen. Then read the program and fill in the missing lines. NOTE: A FOR-NEXT loop with no statement between the FOR and the NEXT statement (see lines 50 and 60) causes a delay by making the computer count to itself. This program pauses after displaying each line.

```
[ ]RUN
COUNTING SLOWLY TO 10

1
2
3
4
5
6
7
8
9
10
```

```
10 _____
20 _____
30 _____
40 _____
50 FOR K = 1 TO 2000
60 NEXT K
70 NEXT X
80 _____
```


6. Look at the output on the screen. Then write a program that will produce this output.

```
[ ]RUN
WHAT IS YOUR LAST NAME?
?GRANT
```

WHAT IS YOUR FIRST NAME?
?KELLY

[illegible][illegible]

 Problem Solving

7. Write a program which allows the user to input any number. Print the user's name as many times as the number he or she inputs.

[illegible]

8. Write a program which uses no more than four lines of BASIC code to print the numbers 1 to 100.

[illegible]

LESSON 14:

More About Loops

■ You can use the STEP function to skip numbers or to count backward.
For example,

```
10 FOR N = 3 TO 21 STEP 3
20 PRINT N
30 NEXT N
```

displays the numbers 3, 6, 9, 12, 15, 18, and 21, one number to a line.
To count backward, add a minus sign to the STEP command. For example,

```
10 FOR X = 20 TO 0 STEP -4
20 PRINT X
30 NEXT X
```

displays 20, 16, 12, 8, 4, and 0, one number to a line.

□□□□□□□□□□□□□□□□□□□□□□□□ **Writing Output** □□□□□□□□□□□□□□□□□□□□□□□□

1. Read the program. Then look at the output and write what is missing on the blank lines.

```
10 PRINT "COUNTING BY 2'S TO 20"
20 PRINT
30 FOR E = 2 TO 20 STEP 2
40 PRINT E;" ";
50 NEXT E
60 PRINT
70 PRINT "COUNTING BACK BY 2'S"
80 PRINT
90 FOR B = 20 TO 2 STEP -2
100 PRINT B;" ";
110 NEXT B
120 END
```

[] RUN
COUNTING BY 2'S TO 20

20, 18, 16, 14, 12, 10, 8, 6, 4, 2

2. Read the program. Then look at the output and write what is missing on the blank lines.

```
10 FOR X = 20 TO 5 STEP -5
20 IF X > 11 THEN GOTO 50
30 PRINT "GOODBYE"
40 GOTO 60
50 PRINT "HELLO"
60 NEXT X
70 END
```

[] RUN
HELLO

3. Read the output on the screen. Then read the program and fill in the missing lines.

```

[ ] RUN
TYPE A NUMBER.
?35
TYPE A LOWER NUMBER.
?30

I WILL COUNT BACKWARDS FROM 35 TO 30.
35 34 33 32 31 30
    
```

```

10 _____
20 _____
30 _____
40 INPUT B
50 PRINT

60 _____
70 FOR X = A TO B STEP -1

80 _____
90 _____
100 _____
    
```

4. Look at the output on the screen. Then read the program and fill in the missing lines.

```

[ ] RUN
0 7 14 21 28 35 42 49 56 63

I AM COUNTING BY:
?9
NO, I AM COUNTING BY 7.
    
```

```

10 FOR B = 0 TO 63 STEP 7

20 _____

30 _____
40 PRINT
50 PRINT "I AM COUNTING BY:"

60 _____
70 IF N = 7 THEN GOTO 100

80 _____

90 _____
100 PRINT "RIGHT!"
110 END
    
```

5. Read the output on the screen. Then write a program that will produce this output.

```

[ ] RUN
TYPE A DECIMAL NUMBER LESS THAN 1.
?0.2
COUNTING FROM 1 TO 2 BY 0.2:
1 1.2 1.4 1.6 1.8 2
    
```

```

10 _____

20 _____

30 _____

40 _____

50 _____

60 _____

70 _____
    
```

6. Read the output on the screen. Then write a program that will produce this output. HINT: Use the formula $A = S * S$ to find the area of a square.

[]RUN

Type the length of a side for the largest square.

?5

Type the length of a side for the smallest square.

3

Side	Area of Square
5	25
4	16
3	9

10 _____

20 _____

30 _____

40 _____

50

60 _____

70 _____

80 _____

90

100 _____

110 _____

120 _____

130 _____

140 _____

Problem Solving

7. Write a program which lets the user input any number from 2 to 10. Have the computer count from 0 to 100 by the number input by the user.

8. Write a program which will allow the user to count backward from 50 to 10 by a decimal number input by the user.

LESSON 15:

Review

Writing Output

1. Read the program. Then look at the output and write what is missing on the blank lines.

```
10 PRINT "WHICH OF THESE CITIES
WOULD YOU MOST LIKE TO VISIT?"
20 PRINT
30 PRINT "1. PARIS"
40 PRINT "2. ROME"
50 PRINT "3. LONDON"
60 PRINT "TYPE IN A NUMBER."
70 INPUT N
80 PRINT
90 IF N = 1 THEN PRINT "YOU WILL HAVE
TO GO TO FRANCE."
100 IF N = 2 THEN PRINT "YOU WILL HAVE
TO GO TO ITALY."
110 IF N = 3 THEN PRINT "YOU WILL HAVE
TO GO TO ENGLAND."
120 END
```

```
[ ]RUN
WHICH OF THESE CITIES WOULD YOU
MOST LIKE TO VISIT?
1. PARIS
```

?2

Writing Input

2. Look at the output on the screen. Then read the program and fill in the missing lines.

```
[ ]RUN
WHAT IS THE NAME OF YOUR BEST FRIEND?
?SARAH
```

```
HOW OLD IS YOUR FRIEND?
?12
```

```
I WILL PRINT YOUR FRIEND'S NAME 12
TIMES:
SARAH SARAH SARAH SARAH
SARAH SARAH SARAH SARAH
SARAH SARAH SARAH SARAH
```

```
10 DIM N$(20)
```

```
20
```

```
30 INPUT N$
```

```
40 PRINT
```

```
50
```

```
60
```

```
70
```

```
80 PRINT "I WILL PRINT YOUR FRIEND'S
NAME" ;X; " TIMES."
```

```
90
```

```
100
```

```
110
```

```
120 END
```

3. Read the output on the screen. Then write a program that will produce this output.

[] RUN
NUMBER FUN

HOW MANY DWARFS DID SNOW WHITE VISIT?

6

NO. SHE VISITED SEVEN DWARFS.

HOW MANY BEARS LIVED IN THE HOUSE
GOLDILOCKS VISITED?

3

YES, THAT'S RIGHT!

10 _____

20 _____

30 _____

40 _____

50 _____

60 _____

70 _____

80 _____

90 _____

100 _____

110 _____

120 _____

130 _____

140 _____

150 _____

160 _____

Problem Solving

4. Write a program which allows a user three guesses to find the square of a number from 1 to 10. Display a reward message if the user's answer is correct.

5. Write a program which displays three random numbers and asks the user if he or she would prefer to see the sum or the product of the three numbers. Allow the user to repeat this process with different numbers as many times as he or she wishes. If the user does not want to run the program again, display the word GOODBYE.

LESSON 16:

Using READ-DATA Statements

■ You can use the READ-DATA statements to tell the computer to read and store a list of data. Whenever the computer sees the READ statement, it searches for DATA statements somewhere in the program. Remember to DIMension string variables. For example,

```
10 DIM X$(10), Y$(10)
20 READ X$, Y$
90 DATA ONE, UNO, TWO, DOS, THREE, TRES
```

instructs the computer to READ the data found in line 90, two pieces at a time, and to store them as X\$ and Y\$. The computer will read as many pieces of data as instructed. Each piece of data is separated by a comma.

Writing Output

1. Read the program. Then look at the output and write what is missing on the blank lines.
NOTE: STOP is used to end the program.

```
10 DIM N$(10)
20 PRINT "NAME", "MATH", "SCIENCE",
"ENGLISH"
30 READ N$,M,S,E
40 IF N$ = "STOP" THEN END
50 PRINT N$,M,S,E
60 GOTO 30
70 DATA ED,90,88,89,SUE,80,86
80 DATA 90,PAUL,78,90,90,FRAN
90 DATA 87,89,89, STOP,0,0
```

[]RUN

NAME	MATH	SCIENCE	ENGLISH
_____	_____	_____	_____
_____	_____	_____	_____
FRAN	87	89	89

Writing Output

2. Look at the output on the screen. Then read the program and fill in the missing lines.
NOTE: A FOR-NEXT loop is used to end the program after four sets of data are read.

[]RUN		
AREA OF TRIANGLES		
BASE	HEIGHT	AREA
10	5	25
6	4	12
7	4	14
8	4	16

```
10 _____
20 PRINT
30 _____
40 PRINT "_____"
50 FOR N = 1 TO 4
60 _____
70 PRINT B, H, (B*H)/2
80 _____
90 DATA 10, 5, 6, 4, 7, 4, 8, 4
```

3. Look at the output on the screen. Then read the program and fill in the missing lines.
NOTE: Remember that DATA can appear anywhere in a program.

```
[ ]RUN
THE OPPOSITE OF HOT IS COLD.
THE OPPOSITE OF LIGHT IS DARK.
THE OPPOSITE OF NEAR IS FAR.
THE OPPOSITE OF LARGE IS SMALL.
THE OPPOSITE OF IN IS OUT.
```

```
10 DATA HOT,COLD,LIGHT,DARK
20 DATA NEAR,FAR,LARGE,SMALL
30 DATA IN,OUT
```

40 _____

50 _____

60 READ A\$, B\$

70 _____

80 NEXT X

4. Look at the output on the screen. Then read the program and fill in the missing lines.
NOTE: Use $(2*L) + (2*W)$ to find the perimeter.

```
[ ]RUN
LENGTH      WIDTH      PERIMETER
-----
3            4            14
6            7            26
5            3            16
2            10           24
8            2            20
```

```
10 PRINT "LENGTH", "WIDTH", "PERIME-
TER"
```

20 _____

30 _____

40 _____

50 _____

60 NEXT P

70 DATA 3, 4, 6, 7, 5, 3, 2, 10

80 DATA 8, 2

5. Read the output on the screen. Then write a program that will produce this output.

```
[ ]RUN
GRADE 6      GRADE 7      GRADE 8
-----
SARAH        BETH          GLORIA
EDWARD       PETER         JEFF
AMY          MARIA         JENNY
JASON        CHRIS         FRED
```

10 _____

20 _____

30 _____

40 _____

50 _____

60 _____

70 _____

80 _____

90 _____

100 _____

LESSON 17:

Producing Low-Resolution Graphics

■ You can use graphics commands to display pictures in color on your screen. For example, when you keyboard

```
10 GR. 3
20 SETCOLOR, 0, 12, 1
30 COLOR 1
40 PLOT 10, 5
```

you are telling the computer the following: Line 10 sets the G^Raphics mode to GR.3. Most of the screen becomes a graph with 40 small squares across and 20 down that can be colored in to make a pattern. Line 20 tells the computer that you will select a color for the graphics squares (0), that you have chosen green (12), and that you want it to be very bright (1). The first number after SETCOLOR chooses what you want to color. The second chooses the color. There are sixteen colors to choose from:

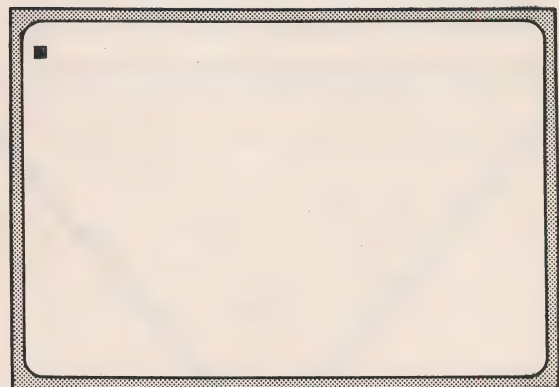
0 gray	6 lavender	11 green-blue
1 gold	7 blue	12 green
2 orange	8 blue	13 yellow-green
3 red-orange	9 light blue	14 orange-green
4 pink	10 turquoise	15 light orange
5 purple		

■ The third number after SETCOLOR chooses the brightness. You can use the numbers 0 (brightest) to 14 (darkest). The COLOR statement in line 30 puts the color on the screen. This number must always be one more than the first number in the SETCOLOR statement. Line 40 lights up one square at column 10 across and row 5 down.

Writing Output

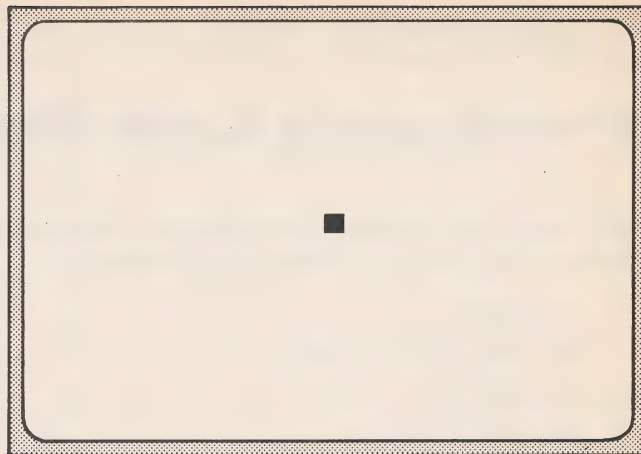
1. Read the program. Then look at the sample screen and draw the missing output. NOTE: GR. 0 returns the screen to the normal text screen.

```
10 GR. 3
20 SETCOLOR 0, 4, 1
30 COLOR 1
40 PLOT 0,0
50 PLOT 39, 0
60 PLOT 20, 19
70 FOR K = 1 TO 1000
80 NEXT K
90 GR. 0
100 END
```



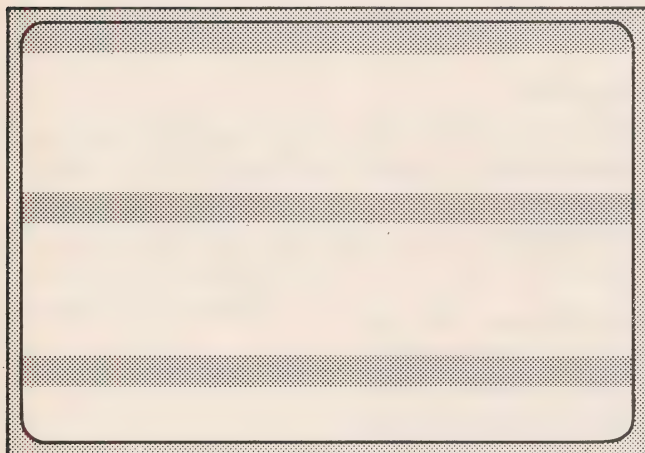
2. Read the program. Then look at the sample screen and draw the missing output. NOTE: The DRAWTO statement will draw a line from the point plotted to the coordinates given in the DRAWTO statement.

```
10 GR.3
20 SETCOLOR 0,10,1
30 COLOR 1
40 PLOT 0,10
50 DRAWTO 39,10
60 END
```



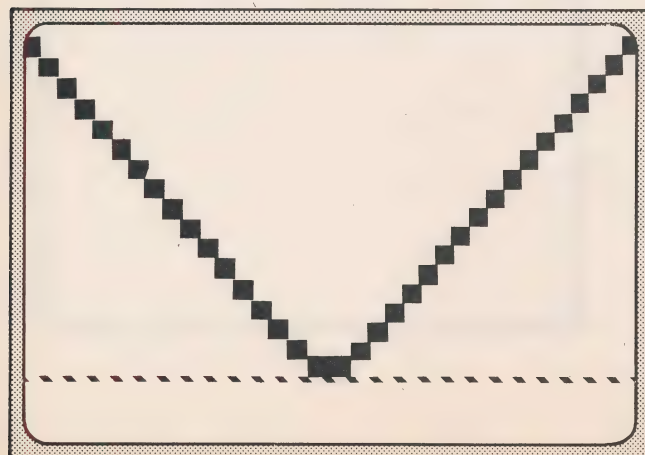
Writing Input

3. Look at the output on the screen. Then read the program and fill in the missing lines.



```
10 GR.3
20 SETCOLOR 0,2,2
30 _____
40 PLOT 0,0: DRAWTO 39,0
50 _____
60 PLOT 0,19: DRAWTO 39,19
70 _____
```

4. Look at the output on the screen. Then read the program and fill in the missing lines.



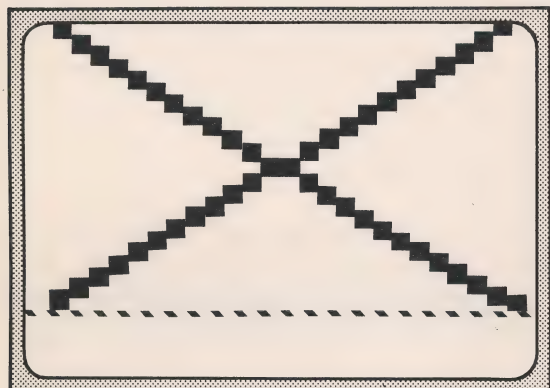
```
10 _____
20 SETCOLOR 0,5,4: COLOR 1
30 PLOT 0,0
40 _____
50 PLOT 20,19
60 _____
70 END
```

5. Look at the output on the screen. Then read the program and fill in the missing lines.
NOTE: GR. 1 allows you to write on the screen in a special graphics text. Use the command PRINT #6; "your message". It is not necessary to choose a color.



```
10 GR.1
20 PRINT #6; "HELLO"
30 _____
40 _____
50 _____
60 _____
70 END
```

6. Look at the output on the screen. Then write a program that will produce this output.



```
10 _____
20 _____
30 _____
40 _____
50 _____
60 _____
```

Problem Solving

7. Write a program which will color the Atari screen blue. Use a FOR-NEXT loop to do this.

8. Write a program which will display your first, middle, and last name on the graphics screen. Print each of the three names on a different line. Then print all three on the fourth line.

Animating Graphics

■ You can use FOR—NEXT loops to move, or animate, shapes across the screen. For example,

```
20 FOR X = 10 TO 20
30 SETCOLOR 0, 5, 6
40 COLOR 1: PLOT X,0
50 COLOR 0: PLOT X,0
60 NEXT X
```

will plot a purple square and then erase it. COLOR 0 without a SET-COLOR statement changes the color of the square to the background color.

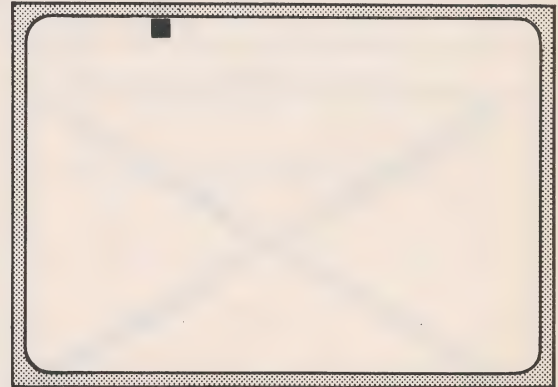
Writing Output

1. Read the program. The sample screen shows the starting position of the square. Draw a square at the ending position. HINT: The delay loop in line 60 slows down the speed of the moving point.

```

10 GR.3
20 FOR Y = 0 TO 19
30 SETCOLOR 0, 7, 4
40 COLOR 1
50 PLOT 10,Y
60 FOR K = 1 TO 50:NEXT K
70 COLOR 0
80 PLOT 10,Y
90 NEXT Y
100 END

```

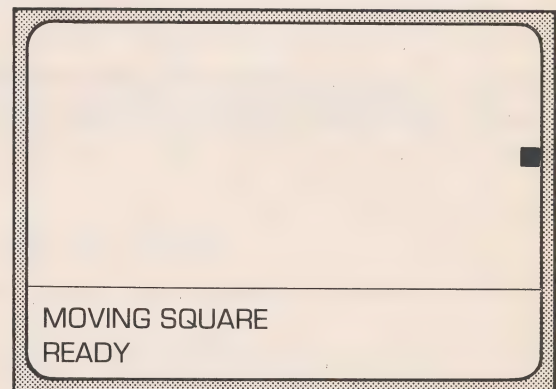


2. Read the program. The sample screen shows the starting position of the square. Draw a square at the ending position. NOTE: The GR.3 command leaves the bottom four lines of the screen for text.

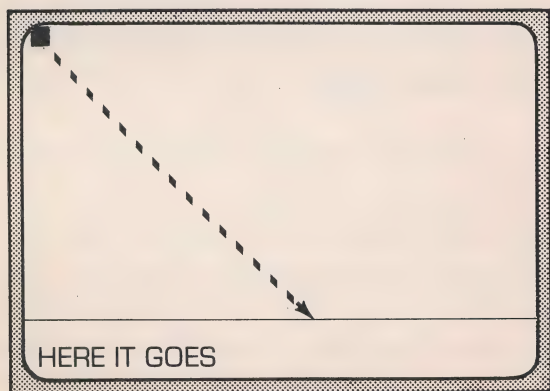
```

10 GR.3
20 PRINT "MOVING SQUARE"
30 FOR X = 38 TO 1 STEP -1
40 SETCOLOR 0, 13, 4
50 COLOR 1
60 PLOT X,10
70 FOR K = 1 TO 50:NEXT K
80 COLOR 0
90 PLOT X,10
100 NEXT X
110 END

```



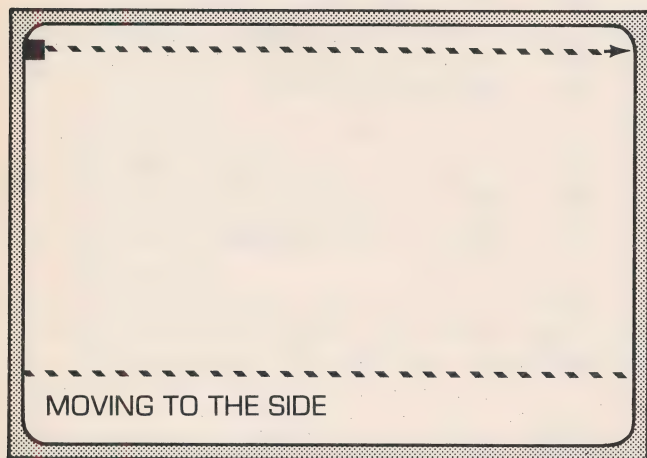
4. Look at the output on the screen. Then read the program and fill in the missing lines.



110 END

160 END

5. Look at the output on the screen. Then write a program that will produce this output. HINT: Don't forget to slow down the square's movement with a delay loop.

[illegible]

Problem Solving

6. Write a program that will move a horizontal line from the top to the bottom of your screen. Make the line light blue. Remember to slow down the line's movement.

[illegible]

7. Write a program that displays a picture of your choice and animates part of it. Remember, all animation is done in the same way: plot, erase, replot.

[illegible]

LESSON 19:

Using String Functions

■ String functions allow you to manipulate string variables in a variety of ways. For example,

```
10 LET A$ = "COMPUTERS ARE FUN"  
20 PRINT A$ [15]
```

will direct the computer to print the substring FUN, which begins at the 15th position of A\$. When only one number is indicated in the substring statement, the computer will print all the characters from that position to the end of the string.

□□□□□□□□□□□□□□□□□□□□□□□□ Writing Output □□□□□□□□□□□□□□□□□□□□□□□□

1. Read the program. Then look at the output and write what is missing on the blank lines.

```
10 DIM A$(20), B$(20)  
20 LET A$ = "MY ATARI IS SMART."  
30 PRINT A$  
40 LET B$ = "BUT I AM SMARTER."  
50 PRINT B$  
60 PRINT  
70 PRINT A$(4)  
80 PRINT B$(5)  
90 END
```

[]RUN

BUT I AM SMARTER.

-
2. Read the program. Then look at the output and write what is missing on the blank lines.
NOTE: Using two numbers in a substring statement indicates the beginning and ending positions of the substring.

```
10 DIM X$(20), Y$(20)  
20 LET X$ = "I LIKE PROGRAMMING."  
30 LET Y$ = "IT IS EASY TO DO."  
40 PRINT X$(8,18), Y$(4,10)  
50 PRINT  
60 PRINT X$  
70 PRINT Y$  
80 END
```

[]RUN

IT IS EASY TO DO.

3. Look at the output on the screen. Then read the program and fill in the missing lines.

```
[ ]RUN
COMPUTERS ARE FUN.
THEY CAN COMPUTE LARGE SUMS.

COMPUTERS    CAN    COMPUTE
```

```
10 DIM P$(20), Q$(28)
20 LET P$ = "COMPUTERS ARE FUN."
30 LET Q$ = "THEY CAN COMPUTE
LARGE SUMS."
```

```
40 _____
50 _____
60 _____
70 _____
80 END
```

4. Look at the output on the screen. Then read the program and fill in the missing lines.

```
[ ]RUN
BULLETIN

BULL
LET
IN
TIN
```

```
10 DIM Z$(10)
```

```
20 _____
30 PRINT Z$
40 _____
50 _____
60 _____
70 PRINT Z$(7,8)
80 _____
90 _____
```

5. Read the output on the screen. Then write a program that will produce this output.

```
[ ]RUN
DIMENSION
_____
DIM
MEN
DIME
ON
```

```
10 _____
20 _____
30 _____
40 _____
50 _____
60 _____
70 _____
80 _____
90 _____
```


LESSON 20:

Review

Writing Output

1. Read the program. Then look at the output and write what is missing on the blank lines.

```
10 DIM A$(5), B$(5), C$(5), D$(5)
20 DATA 1) *,2) #,3) $,4) ?,4
30 READ A$, B$, C$, D$, E
40 PRINT "AFTER AN INPUT STATEMENT,"
50 PRINT " THE COMPUTER DISPLAYS A "
60 PRINT A$
70 PRINT B$
80 PRINT C$
90 PRINT D$
100 PRINT
110 PRINT "TYPE A NUMBER"
120 INPUT R
130 IF R < > E THEN GOTO 160
140 PRINT "GOOD WORK!"
150 GOTO 170
160 PRINT "THE COMPUTER DISPLAYS A ?"
170 END
```

[]RUN

THE COMPUTER DISPLAYS A

3) \$

TYPE A NUMBER

?2

Writing Input

2. Look at the output on the screen. Then read the program and fill in the missing lines.

```
[ ]RUN
20 + 13 =
?33
GREAT!
10 + 17 =
?18
THAT'S NOT IT. TRY AGAIN.
10 + 17 =
?27
GREAT!
14 + 12 =
?26
GREAT!
```

```
10 FOR A = 1 TO 3
20 LET X = INT(20 * RND(1)) + 1
30
40 PRINT X; " + "; Y; " = "
50
60 IF R < > X + Y THEN GOTO 90
70
80
90
100 GOTO 40
110 NEXT A
120 END
```

3. Read the output on the screen. Then write a program that will produce this output.

[] RUN
THE PYRAMIDS ARE LOCATED IN
1. ITALY
2. SPAIN
3. EGYPT

TYPE A NUMBER.
??
NO. THE PYRAMIDS ARE IN EGYPT.

This image shows a single page from a lined notebook. The page features horizontal blue ruling lines spaced evenly down its length. On the left side, the numbers 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, and 130 are printed vertically, corresponding to each set of lines. The paper has a slightly off-white or cream color, and there are no markings or writing on the page itself.

Problem Solving

4. Write a program which displays a graphic of your choice and animates part of it. Print a title for your graphic in the text portion of the graphics screen.

[illegible]

5. Write a program which asks the user multiple-choice questions about a book you read in class. If the user's answer is correct, display a reward message. If his or her answer is incorrect, supply the correct answer.

Summary of BASIC Statements and Commands for the ATARI

Statement	Example	Result
INPUT/OUTPUT STATEMENTS		
PRINT	PRINT "HI"	Displays the word HI on the CRT.
	PRINT 2 + 3	Displays 5, the value of 2 + 3.
	PRINT	Displays a blank line.
INPUT	INPUT X	Displays a ? and waits for the user to type a value for X.
LET	LET X = 7	Assigns the value 7 to the variable X.
READ-DATA	READ Z: DATA 1,2	The computer READs the values in the DATA statement and stores them as Z.

FLOW OF CONTROL STATEMENTS

IF-THEN	IF A = 1 THEN GOTO 20	If the value of A = 1, the computer will branch to line 20. If A < > 1, the next line in the program will be executed.
GOTO	GOTO 50	Directs the computer to line 50.
FOR-NEXT	FOR P = 1 TO 5:NEXT P	Instructs the computer to repeat the statements between the FOR and the NEXT statement five times.

GRAPHICS STATEMENTS

GR.O	GR.O	Sets the screen to the normal text mode. Use this to return to the text screen from a graphics program.
GR.3	GR.3	Sets a graphics screen with 40 squares horizontally and 20 vertically.
GR.1	GR.1	The graphics mode which will display a special graphics text on the screen.
SETCOLOR	SETCOLOR 0,12,4	Lets the computer know that the programmer will choose the color for the graphics squares, it is green, and the brightness level is 4 [medium].
COLOR	COLOR 1	Places the color on the screen. The number must always be 1 more than the first number in SET-COLOR.
PLOT	PLOT 3,5	Plots a square at 3 across and 5 down.
DRAWTO	DRAWTO 10,15	Draws a line from the point plotted to 10 across and 15 down.

FORMAT STATEMENTS

comma	PRINT 1,2,3,4	Directs the computer to print the numbers 1, 2, 3, and 4 tabbed across the screen at columns 0, 10, 20, and 30.
semicolon	PRINT "HI ";N\$	Directs the computer to print the value of N\$ next to the word HI on the same line of the screen.

FUNCTIONS

RND [1]	PRINT RND [1]	Prints a random decimal number between 0 and 1.
INT	PRINT INT [23.4]	Prints 24, the whole number without the decimal.
INT [10*RND[1]]	PRINT INT [10*RND[1]]	Prints a number between 0 and 9.
INT [10*RND[1]]+1	PRINT INT [10*RND[1]]+1	Prints a number between 1 and 10.

ARITHMETIC AND LOGICAL SYMBOLS

+	PRINT 3 + 3	Directs the computer to add.
-	PRINT 3 - 3	Directs the computer to subtract.
*	PRINT 3 * 3	Directs the computer to multiply.
/	PRINT 3 / 3	Directs the computer to divide.
^	PRINT 3 ^ 2	Directs the computer to square 3.
<	A < B	A is less than B.
>	C > D	C is greater than D.
=	X = Y	X is equal to Y.
<>	P <> Q	P is not equal to Q.
<=	K <= H	K is less than or equal to H.
>=	R >= S	R is greater than or equal to S.

SYSTEM COMMANDS

NEW	NEW	Erases the current program from memory.
LIST	LIST	Lists the statements of a program.
SAVE	SAVE FIG	Saves the program named FIG.
RUN	RUN	Directs the computer to execute the program.
LOAD	LOAD MIN	Loads the program named MIN into memory.
CONT	CONT	Continues the program after BREAK is pressed, or after the computer meets a STOP or END in the program.

GLOSSARY

BASIC the language used by the ATARI computer. The word BASIC stands for Beginner's All-purpose Symbolic Instruction Code.

cassette recorder an input/output device used to store information on a cassette tape or to load information from a tape into the computer's memory.

cassette tape a secondary storage device for the ATARI. Information is stored on magnetic tape.

CRT Cathode Ray Tube. An output device such as a television or monitor.

disk drive an input/output device used to store information on a disk or to load information from a disk into the computer's memory.

floppy disk a secondary storage device for the computer. Information is stored on magnetic tape.

hard copy the printed results of a computer program.

input any information typed into the computer or loaded in from a disk or a cassette tape.

output the results of running a computer program. The output may appear on a monitor or be printed on a printer.

printer an output device which produces the hard copy of a program.

program a list of instructions which tells the computer to perform a certain task.

RAM Random Access Memory. A temporary memory used to store data or programs inside the computer. Anything stored in RAM is erased when the ATARI is turned off.

ROM Read Only Memory. The ATARI's permanent memory. This contains the permanent information written to make the computer operate.

software programs which have been stored on tape or on a disk.

string any letters, characters, or numbers which will not have arithmetic operations performed on them.

statement a word or phrase in BASIC which is used to give the computer an instruction.

DISK COMMANDS FOR THE ATARI

SAVING A PROGRAM ON A DISK:

When you have finished typing your program, type the word LIST and press RETURN to make sure that there are no errors in your program. Place an initialized disk in Drive 1, and type the word SAVE followed by quotation marks, a capital D, a colon, and the name of your program. If you wish to save a program named MATH DRILL, you would type the following:

SAVE "D:MATH DRILL"

You may choose any name you wish, but the name may not begin with a number and may not contain a comma. Press RETURN and you

will see the BUSY light on the disk drive go on. Do not open the door of the drive until the light is off. Once the light is off and the READY message appears on the screen, your program has been saved on the disk.

LOADING A PROGRAM FROM A DISK:

Type the word DIRECTORY and press RETURN to see the contents of the disk. After you have selected the program, type LOAD followed by the name of the program in quotation marks. You can RUN the program after the BUSY light on the disk drive has gone out.

TAPE COMMANDS FOR THE ATARI

SAVING A PROGRAM ON TAPE:

When you have finished typing your program, LIST and RUN it to be sure it does not contain errors. Then place a cassette tape in your recorder and rewind it to the beginning. Type the command CSAVE. You will then hear two bells which indicate that you should press the PLAY and RECORD buttons on your tape recorder. Then press RETURN. When the recording is finished, you will hear another bell. Press the STOP button on your tape recorder. Your program is now saved on a cassette tape.

LOADING A PROGRAM FROM A TAPE:

Rewind the tape to the beginning and type the command CLOAD. As soon as you hear a bell, press the PLAY button on the tape recorder, followed by pressing RETURN. When all the information is loaded into the computer, you will see the READY message on your screen. Press the STOP button and RUN the program.

Atari Graphics Sheet for GRAPHICS 3

Across

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38

Down
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

